In the Specification

Please replace the paragraph beginning at Page 5, line 1 with the following:

While [his] this address switching scheme provided maintenance and load balancing functions and enhancements, its implementation prevented any ready use with groups of logical volumes without difficulty. From a practical standpoint it was also not possible to respond to certain interruptions as might be caused by a problem with a specific physical drive or by failure of a connecting channel.

Please replace the paragraph beginning at Page 5, line 23 with the following:

This particular application of address switching is involved in I/O and error recovery programs. It allows the host with an error protection program to perform diagnosis and provide control for a single device. Like the modified system of United States Letters Patent No. 6,092,066, the program is directed primarily to swapping one logical volume at a time. Swapping multiple logical volumes is not proposed. Thus if I/O requests from a host are directed to several logical volumes, each time an error is recognized on one logical volume, that transfer and resulting address switching occurs only with

respect to that logical volume. None of the other logical volumes are switched. Consequently the process must repeat each time to divert I/O requests to related [logical volumes to a] redundant logical [volume] <u>volumes</u>. What is needed is a method and apparatus for enabling a host at one location to redirect I/O requests to a data storage facility at another location that relate to a plurality of logical volumes and to enable such transfers to made either at operator discretion or automatically in response to certain failures.

Please replace the paragraph beginning at page 9, line 14 with the following:

By way of background for understanding this invention, FIG. 1 depicts a data processing network or arrangement 200 with systems at geographically diverse sites, namely: a local system <u>site</u> 201 and a remote system <u>site</u> 202. The local site 201 includes a local host 203 with a local storage controller or data storage facility 204. As known, the local host 203 communicates with the local data storage facility 204 through a channel 205. Operating systems for providing such communications including processor I/O requests are well known in the art. Similarly, the remote site 202 includes an optional remote host 206 with the capability of directing I/O requests to a remote data storage facility 207 through a

channel 208. Each of the local and remote data storage facilities 204 and 207 comprises a disk array storage device.

Please replace the paragraphs beginning at page 11, line 6 through page 12, line 2 with the following:

The system 200 shown in FIG. 1 includes a remote adapter 230 in the local data storage [device] facility 204 and a remote adapter 231 in the remote data storage [device] facility 207 interconnected with a separate communications path 232. This communications path 232, sometimes called a COMM link, is a high bandwidth path that may extend for up to thousand of miles and incorporate high-speed data transmission equipment, as known.

During normal mirroring operations an I/O request directed to the local data storage facility may request data to be stored in a logical device. The data is stored initially in the system memory 212. That data destages to an appropriate one of the physical storage devices 214 or 216 through the respective disk controllers 213 and 215. In addition, the remote adapter [232] 230 transfers the data from the system memory 212 over the link 232 through the remote adapter 231 into the system memory 222. The remote data storage device 207 then destages this data to the corresponding location in one of the sets of physical storage devices 224 and 226 through the

device controller 223 and 225, respectively. The remote adapters 230 and 231 and communications link 232 can also be configured to transfer data from the remote data storage facility 207 to the local data storage facility 204.

Please replace the paragraphs beginning at page 12, line 17 through page 13, line 22 with the following:

The hosts 203 and 206 shown in FIG. 1 may operate with any commercially available operating system, such as the [IBM] MVS operating system <u>of International Business Machines Corporation</u>. In an MVS environment, a host will include a plurality of CPs or central processors. By way of example, FIG. 2 depicts two central processors CP(1) and CP(n) associated with the local host 203 and identified by reference numerals 237 and 238, respectively. The remote host 206 normally will a similar structure. The CPs communicate with a main storage unit, such as a main storage unit 239 in the local host 203. As known, the main storage unit 239 is divided into a number of areas including common and private areas and extended common and extended private areas that store data and programs, or application, such as an APPL application 240 and an AUTO SDAS application 241 that implements this invention. A console device 242 permits an operator to communicate with the system for performing a number of configuration, diagnostic and

other procedures independently of operations that occur in response to any application programs. As will become apparent, an operator can use the console device 242 to initiate switching in accordance with one aspect of this invention.

A control block, called a Unit Control Block (UCB) in the context of an MVS operating system, contains information about a logical volume (hereinafter a "logical device") and related status information. Two blocks are shown in FIG. 2; namely a UCB(R1) unit control block 243 and a UCB(R2) unit control block 244. The common address space will contain an additional [UCB] <u>unit control block</u> corresponding to the UCB(R1) control block 243 for each logical device in the local host 203. It will also contain a unit control block (UCB) like the UCB(R2) control block 244 corresponding to each logical device associated with the remote host 206.

Please replace the paragraph beginning at page 14, line 6 with the following:

A load and initialize module (not shown) establishes the loads the functional modules 241 into the address space 239. Such a module may also define a data structure, such as the data [structures] <u>structure</u> 245. FIG. 3 represents this data structure as a two-dimensional matrix in which each row

represents a different logical device, designated as DN1, DN2, DN3 and DNn logical devices.

Please replace the paragraph beginning at page 15, line 11 with the following:

A swap device module 256 performs a swapping operation for a particular device. If, for example, the DN1 logical device is identified, the swap device module 256 exchanges the information in the corresponding UCB(R1) and UCB(R2) unit control blocks as identified by the information in the corresponding UCB PTR(R1) and UCB PTR(R2) pointers 245 and 255. A swap group module 257 enables all the logical devices in a group established by a Define_Group command module 249 to be switched. A swap interface module 258 presents alternate approaches for instituting the processes of swap group module 257 and/or swap device modules 256.

Please replace the paragraphs beginning at page 16, line 12 through page 18, line 6 with the following:

Step 261 determines whether the Define_Group command arguments requested a validation procedure. If they do, step 262 calls a Validate Group process as described with respect to FIG. [5] 6. The Validate Group process uses the information in the command and other sources, such as the RDF configuration

files that are part of the system shown in FIG. 1, to generate information for the data structure 245. This information includes the previously described source and target identifications, group validity state, the operating mode and the pointers to each UCB.When the Validate Group process completes, control passes to step 263. If all the logical devices in the group and other related tests are valid, control passes to step 264. Step 264 determines whether the Define_Group command arguments request immediate dynamic address switching, or "swap". The "swap" can be limited to a single logical device or to all logical devices in a group. If a swap is defined for a group, step 264 transfers to step 265 that initiates the Swap Group module 257 in FIG. 3 as also shown in FIG. 9. When the Swap Group procedure is complete, control returns to FIG. 4 and particularly to step 266 that processes an appropriate status message for transfer back to the application program as known in the art. If the Define_Group Command [and] were processed as just described, such a status message would indicate successful definition of the group and completion of the swap.

If the command is found to be invalid, step 260 transfers control directly to step 266 to produce an appropriate status message indicating that determination. If the Define_Group command validate group does not contain a validation argument,

step 261 transfers control directly to step [266] 263. This test might also be made as part of step 260. If step 263 determines that the group is invalid, control also transfers to step 266. If the Define_Group command does not include a swap argument, step 264 transfers directly to step 266 to generate the appropriate status message.

FIG. 4 uses step 262 to initiate a group validation. In accordance with another aspect of this invention, there are other ways to initiate a validation. Referring now to FIG. 5, the system operator may elect to have the validation module operate asynchronously on a repetitive basis in accordance with another argument in the Define_Group command. Step 270 represents a timing mechanism that would establish such repetitive operations of a call [271] for the validate group process 271. The interval between successive calls could be measured in minutes, hours or longer.

Please replace the paragraph beginning at page 18, line 22 with the following:

A more detailed description of the validate process 246 is depicted in FIG. 6. Step 280 obtains a device specification from the group definition as contained in the data structure 245 shown in FIG. 3. Step 281 selects a next source device from the data structure 245. Step 282 transfers control to

step 283 if the selected source UCB is defined. If it is, step 283 obtains the information about the corresponding target device. If the target UCB is defined, step 284 transfers control to step 285 to determine whether the UCB states are valid. This test, for example, assures that the corresponding UCBs exist at an MVS level[, this test] <u>and</u> can assure that the source and target devices are on-line. Still other tests may be conducted. If both UCBs are valid, step 286 obtains information about the data storage facility and the RDF configuration. For example, step 287 determines whether the validation will be used for performing dynamic address switching from the local system 201 to the remote system 202 in FIG. 1 or the reverse as would occur when it was possible to redirect I/O requests to the local data storage facility 204. If the former condition exists, control passes to step 290. Otherwise step 291 determines whether the R1 device as a target device for this dynamic address switching operation is ready. Other tests might also be used<u>.</u> [to ] If [it] <u>the R1 device</u> is <u>not</u> ready, control transfers to step 290.

Please replace the paragraph beginning at page 20, line 6 with the following:

Assuming that the logical volumes are to operate in such a mode, control transfers to step 292 in FIG. 6 that assures that

both data storage facilities are valid and are operating properly and that corresponding source and target devices are compatible. Step 293 determines whether any software features are installed that might conflict with an address switching operation. If all these tests are completed satisfactorily, the validate group procedure builds a device control extension (DCE) and establishes a path group to the target device in step 294. This path group extends to the R2 logical device in the remote data storage facility in the case of a dynamic address switch from a logical device in the local data storage facility 204. For a reverse dynamic address switching operation, the target device is the logical device in the local data storage facility 204.

Please replace the paragraph beginning at page 25, line 4, with the following:

If the swap is successful, step 325 returns control to step 333 to select a next valid source device. This process will repeat until the information for each set of UCBs has been exchanged. If the swap for a selected source device is not successful, step 325 transfers to step 326. If a consistent swap operation is underway, step 326 transfers control to step 327 that terminates the swapping operation and performs a reverse swapping operation to return all previously swapped

source and target devices to their status prior to the beginning of the consistent address switching operation. When this has been completed, step 330 lowers the IOS level for all the valid devices. If the operation does not involve a consistent swap, step [331] <u>332</u> merely selects a next valid source device. If such a device is available, step 333 transfers control back to step 323 to select a next valid device. When all the address switching is completed for the group by exchanging UCB information, the swap group procedure completes by lowering the IOS levels in step 330. When this occurs after a successful procedure, I/O hosts are directed to the mirror devices such as logical devices in the remote data storage facility over a new channel.